

TECHNICAL REPORT

Data Conversion of the ACOS Magnetic Tapes by Using the Fujitsu Computer System

Shigetoshi TAKATA and Koichi OHTA

Abstract

In our institute, solar-terrestrial observational data have stored in a few thousands of magnetic tapes of the FRC format, which is a binary file format in the NEC ACOS-6 computer system. It became recently necessary to directly handle the data of the ACOS magnetic tapes by using the Fujitsu computer system. The data conversion from the ACOS FRC format to the Fujitsu binary format was, however, formerly necessary a long time and a lot of manpower in the several operation steps because a complicated process had to be done. That is, the ACOS magnetic tapes were once converted to the IBM format tapes due to the ACOS spooling procedure in the ACOS computer and then the IBM format tapes were converted to the Fujitsu binary file in the Fujitsu computer. In this time, we have developed a new program of the data conversion in which the NEC FRC format data files can be directly handled in the Fujitsu computer system. As the result, many observational data can be easily and quickly treated in a IBM compatible computer.

富士通計算機システムでのACOS磁気テープの処理

高田 重利 太田 幸一

要旨： 空電研究所では、日本電気製のACOS-6系汎用コンピュータのファイル形式であるFRC形式で作成されたデータが大量に磁気テープに保存されている。そのデータを富士通製の汎用コンピュータ（FACOM）で一括して扱えるようにする要求が最近高まってきた。しかし、今までその交換には、ACOSシステムのデータ管理サービス用のスプール処理システムを利用して、一度、ACOS側でIBMファイル形式に変換してさらにFACOM側で変換作業を行なうと言う、両システムでの交換を行なっており、時間がかかり、人的操作ステップの多い作業であった。

今回、FACOM側で直接FRC形式のデータテープを処理する、スピードと操作性を良くした処理プログラムを開発したので報告する。

1. はじめに

空電研究所では、1977年から日本電気製のACOS-6系汎用コンピュータが、稼働している。現在まで、11年の間にACOS-6の古い標準ファイル形式であるFRC形式⁽¹⁾⁽²⁾で作成された観測データやシミュレーションデータが、数千本の磁気テープとして蓄積、保存されている。最近、そのデータを富士通製の汎用コンピュータ（FACOM）で一括して扱えるようにする要求が高まってきた。今まで、そのデータをFACOMで利用するときは、次の様な手順が必要であった。先ず、空電研究所で磁気テープから磁気ディスクにコピーし、ACOSのデータ管理サービス用のスプール処理システム⁽³⁾を利用して、IBMファイル形式に変換して磁気テープに書き込む。そして、この磁気テープをFACOMに持って行き、磁気ディスクにコピーして、書式付きで書かれたデータファイルならばそのまま、書式なしで書かれたデータファイルならば内部表現変換を行う。しかし、大量に一括して利用したい場合は、ACOS側ではIBMファイル形式の磁気テープに変換するのに時間がかかり、更に交換作業用の大量の磁気テープが必要となる。一方FACOM側では、磁気ディスクへのコピーで時間がかかり、更にデータの内部表現変換で時間がかかるという効率の悪い変換作業であった。

今回開発したプログラムは、FACOMで動作し、直接ACOSのオリジナルデータテープをFACOMのシステムで処理することができ、処理作業は単純化され、更にACOSからFACOMへの変換が容易となった。

2. ACOS-6系のデータテープの形式

ACOSのファイル形式は、現在はUFAS（ユニファイドファイルアクセスシステム）形式が、古くはFRC（ファイルレコードコントロール）形式が標準で、各形式とも順編成ファイルと直接編成ファイルの2種類がある。順編成ファイルは、複数の物理レコード単位（ブロック）から構成されていて、ブロックは、一つまたは二つ以上の論理レコードとブロック制御語（BCW）から形成されている。ブロックは、320ワード（1ワード36ビット）以下となっている。論理レコードには、固定長と可変長があり、固定長ではレコード制御語（RCW）がなく、可変長ではレコード制御語が付いている。可変長の中で1つのブロック内に納まる長さの論理レコード（318ワード以下）ばかりで形成されている場合を可変長レコード、複数のブロックにまたがる論理レコードがある場合を分割レコードと呼んでいる（図1）。直接編成ファイルは、ブロックが、一つまたは二つ以上のレコードから形成されていて、

ブロックおよびレコードには、制御情報は付加されていない、レコードは、非分割レコードと分割レコードがある。

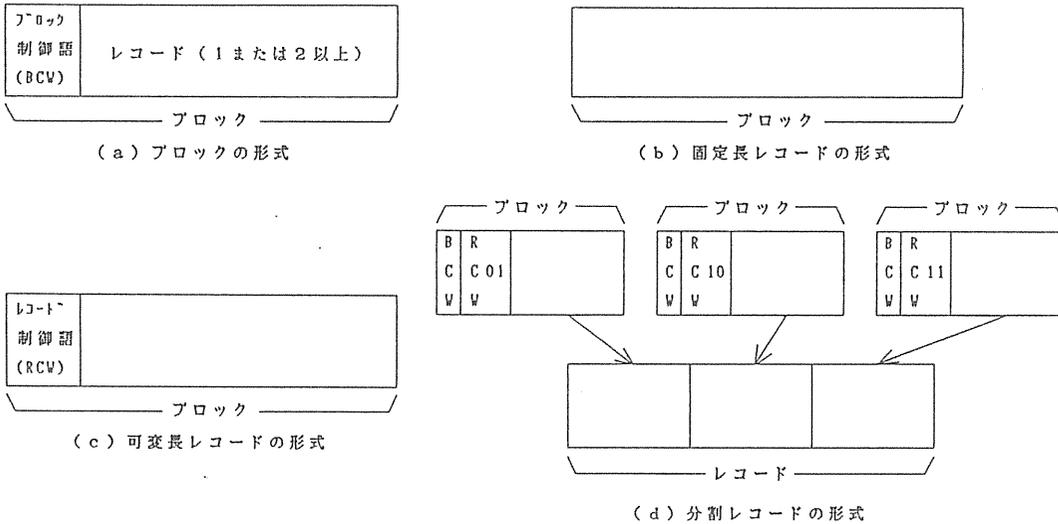


図1 順編成ファイルの形式⁽¹⁾⁽²⁾

データの内部表現⁽²⁾は、語長が36ビットで1バイトが9ビットから構成されていて、2バイト整数型、4バイト整数型、実数型、倍精度実数型、4倍精度実数型、複素数型、倍精度複素数型、4倍精度複素数型、論理型、文字型、日本語型、2進型、8進型がある。

磁気テープのファイル形式⁽¹⁾は、見出しラベル群 (BTL, TM), ファイルデータ (ブロック1-N), 終了ラベル群 (TM, EOF, TM, PTL) から形成されている。複数ファイルボリュームの場合は、見出しラベル (BTL) よりファイル終了ラベル (EOF) の次のテープマーク (TM) までが繰り返されている (図2)。

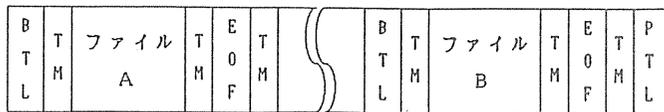


図2 磁気テープファイルの形式⁽¹⁾

データテープには、上記のような内部表現のデータがそのまま、又は外部表現に変換されて書き込まれている。磁気テープへの書き込みは、フォートランプログラムか、ACOSデータ管理サービス用のユーティリティのコピープログラム、又はスプール処理システムのコピープログラムを利用してなされる。

以上の各形式で作成されたデータテープの中で、利用頻度の高い表1の○印の付いている形式のデータテープの変換プログラムを今回作成した。

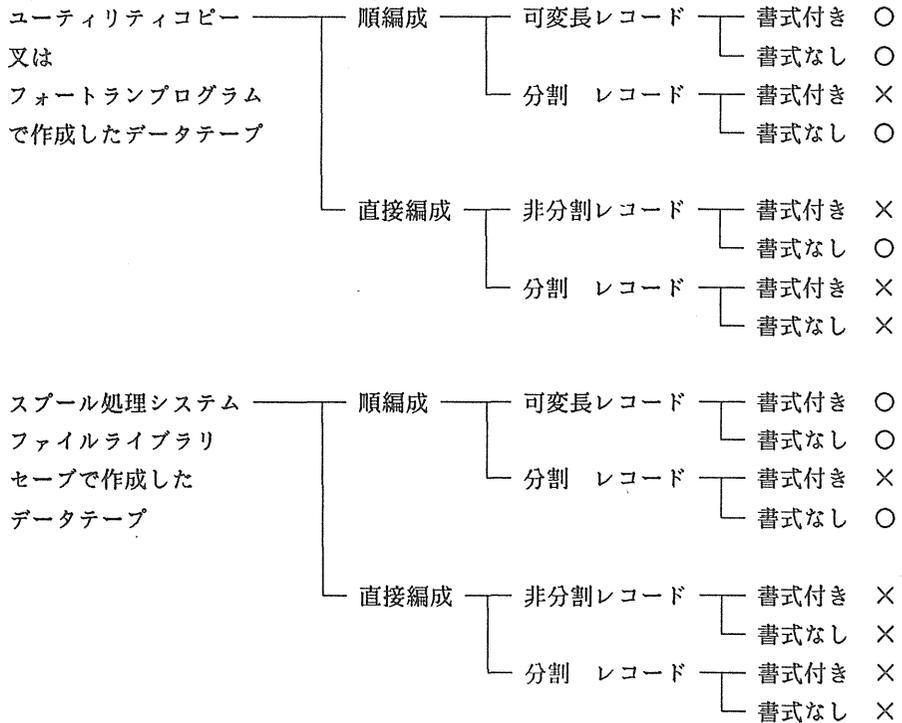


図3 データテープの形式

3. 変換方法と変換プログラムの構成

変換は、次の3段階に分けて、各段階を順番に実行することにより行なわれる。

(1) ブロック長，レコード長，レコード形式を決定する。

決定方法には，データテープの中から取り出したいファイルのフィジカルダンプを取って決定する方法と，フォートランプログラムにより決定する方法とがある。

(2) データテープから磁気ディスクへファイルをフィジカルコピーする。

(3) 変換プログラムを実行する。

以下に各段階の内容を順番に説明する。

3-1 ブロック長，レコード長，レコード形式の決定

3-1-1 フィジカルダンプ

フィジカルダンプ⁽⁵⁾⁽⁶⁾は，リスト1に示すFACOMのユーティリティプログラムMTDUMPを利用して，データテープから取り出したいファイルの順番（ファイル番号）を指定して行う。ファイル番号は，ACOSの磁気テープのファイル形式の見出しラベル（BTL），ファイル終了ラベル（EOF），パーシャルラベル（PTL）も1つのファイルとして数えて指定する。このダンプにより，ブロック長，レコード長，レコード形式を決定する。ブロック長は，全ブロック長の最大長に合わせ，レコード長は，全ブロック長の最大公約数の値に合わせ，レコード形式は，FB形式にする。例えば，あるデータテープを，フィジカルダンプした場合，1440バイト（1バイトは8ビット）の長さのブロックが6個と423バイトの長さのブロックが1個ダンプされた時，ブロック長は最大長の1440バイト，レコード長は1440バイトと423バイトの最大公約数9バイトとする。

```

00010 //A41456A JOB ,CLASS=F
00020 //*
00030 //* MTDUMP MT DUMP PROGRAM
00040 //*
00050 // EXEC MTDUMP,VOL=TOM001,OWNER=NAME,DEN=4,UNIT=FMT6250,TM=2
00060 //

```

リスト1 磁気テープのフィジカルダンププログラム

3-1-2 フォートランプログラム

ブロック長、レコード長、レコード形式の決定は、予めデータテープを作成した時の形式とレコード数と各レコードのデータ数が判っていれば決定できる。このフォートランプログラムは、データテープの形式とレコード数と各レコードのデータ数を与えれば自動的に各数値を決定してくれる（プログラム例1）。このプログラムを作成したことにより、フィジカルダンプを使用するよりも、さらに効率が良くなった。

3-2 フィジカルコピー

コピープログラム⁽⁵⁾⁽⁶⁾は、リスト2に示すFACOMのユーティリティプログラムMTTODAを使用して、磁気ディスクへコピーする。その際、3-1で決定したブロック長、レコード長、レコード形式、ファイル番号、書き込む出力ファイル名を指定する。

```

00010 //A41456A JOB ,CLASS=F
00020 //*
00030 //* MTDK MT COPY PROGRAM
00040 //STEP1 EXEC MTTODA
00050 //INPUT DD UNIT=FMT6250,LABEL=(2,BLP),VOL=SER=99999,
00060 // DCB=(RECFM=FB,LRECL=18,BLKSIZE=1440)
00070 //OUTPUT DD DSN=A41456A.SRW100,DISP=(NEW,CATLG),UNIT=MSS,
00080 // SPACE=(TRK,(400,200),RLSE),VOL=SER=41456C,
00090 // DCB=(RECFM=FB,LRECL=18,BLKSIZE=1440)
00100 //

```

リスト2 磁気テープのフィジカルコピープログラム

3-3 変換プログラム

変換プログラムは、図3に示した各形式に対応して、フォートラン言語⁽⁷⁾⁽⁸⁾で作成されている。プログラムの構成は、図4に示す様になっていて、各構成機能はサブルーチン化されている。プログラムは、扱い易さと変換スピードとプログラムの効率を考慮して作成されている。

以下に各機能の説明をする。

3-3-1 ブロックの読み込み

ブロックの読み込みは、1ブロック単位でデータを読み込んで処理して行くものである。レコードは、

3-1で与えられたレコード長単位で処理される。1レコードは、1バイト単位で1ワードに左詰めに入れて読み込まれる。ブロックは、先ず数レコードが読み込まれ、ACOSのFRC形式のブロック制御語（BCW）のブロック長が調べられる。その長さに合わせてブロックは、読み込まれるので、可変長でも自動的に読み込むことができる。

3-3-2 36ビット変換

36ビット変換は、ACOSの語長36ビットと、FACOMの語長32ビットでは、4ビットのズレが起り、変換作業を行なう場合、そのままのビットの並びではプログラムが複雑となるために、予めACOSの1ワードを取り出すためのものである。先ず、1ワードに1バイト左詰めに読み込んだデータを9バイト持ってきて、ACOSの1ワードを、バッファの2ワードに右詰めまたは左詰めで入れて、ACOSの2ワードを作る。詰め方を変えるのは、変換スピードを上げるためである。図5に右詰めの例を示す。

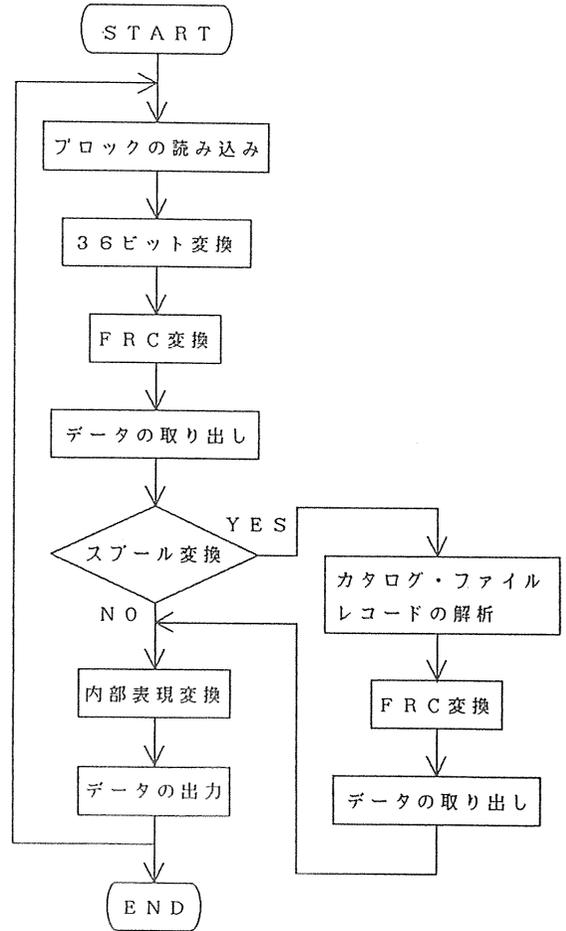


図4 変換プログラムの構成

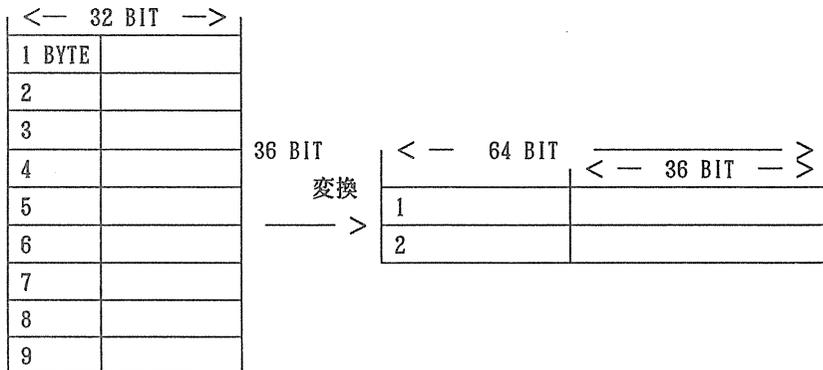


図5 36ビット変換

3-3-3 FRC変換

ACOSのFRC形式は、各ブロックとレコードの1語目に、ブロック制御語（BCW）とレコード

制御語 (RCW) が付加されていて、ブロックとレコードの制御を行なっているファイル形式である。FRC変換は、FRC形式の制御語を解読するものである。制御語の内容は、以下の様になっている。

A. ブロック制御語 (BCW)

ビット 0-17 ブロック通し番号
 18-35 ブロック長

B. レコード制御語 (RCW)

a. 可変長レコードの場合

ビット 0-17 レコード長
 ビット 18-19 文字データの場合、最後の語の使用されている文字数 00 (4文字)
 01 (1文字) 10 (2文字) 11 (3文字)
 ビット 20-23 ビット0-17がゼロのとき、特定のファイルマーク文字が入る
 ビット 24-25 ゼロ
 ビット 26-29 媒体コード (レコード形式)
 ビット 30-35 レポートコード (特定のレポート又はカードデッキを示す)

b. 分割レコードの場合

①最初のセグメント

ビット 0-17 レコード長
 ビット 18-23 使用されない
 ビット 24-25 最初のセグメントであることを示すレコード (01)
 ビット 26-29 媒体コード
 ビット 30-35 レポートコード

②中間セグメント

ビット 0-17 レコード長
 ビット 18-23 使用されない
 ビット 24-25 中間セグメントであることを示すレコード (10)
 ビット 26-35 セグメント番号 (2から始まる)

③最後のセグメント

ビット 0-19 レコード長および最後の語句の使用されている文字数
 ビット 20-23 使用されない
 ビット 24-25 最後のセグメントであることを示すレコード (11)
 ビット 26-35 セグメント番号

3-3-4 データの取り出し

データの取り出しは、FRC変換によって取り出されたブロック長、レコード長から可変長レコードまたは分割レコードに合わせてレコードを取り出し、レコードの中からデータを取り出すものである。

3-3-5 スプール変換

スプール変換は、ACOSスプール処理システムのファイルライブラリテープ作成機能でコピーされたデータテープを変換するものである。データテープには、磁気ディスク内でのカタログ・ファイル記述レコードが付加されて、図6に示す様に、磁気ディスクファイルのイメージがそのままコピーされて

いる。この形式でコピーされたデータテープは、スプール変換した後もう一度FRC変換をしてデータを取り出さなければならない。スプール処理システムの中の情報交換テープ作成機能で作成されたFRC形式のデータテープは、ユーティリティのコピーの場合と同じで、磁気ディスクファイルがそのままコピーされていて、直接編成ファイルには、ブロック制御語（BCW）が付加されてコピーされている。

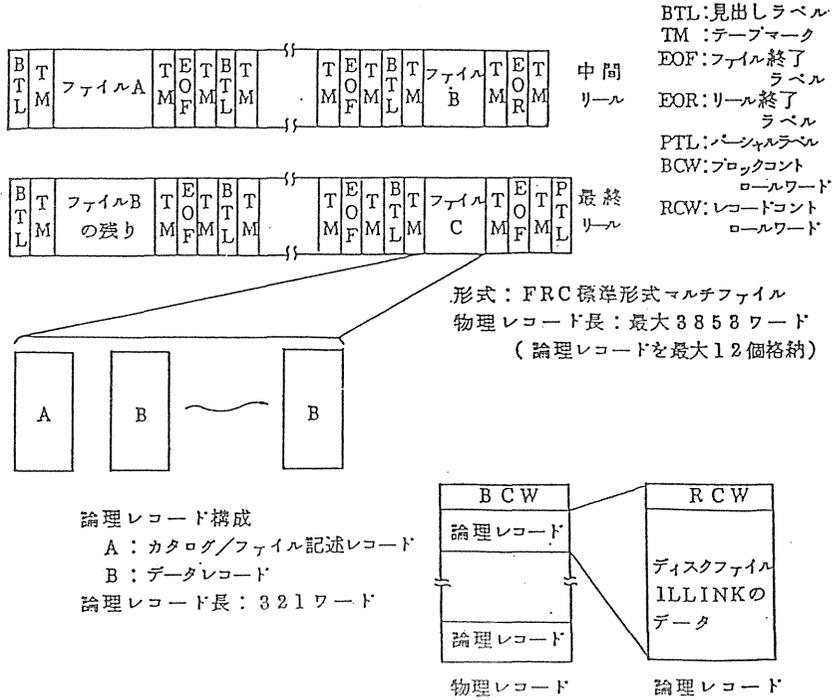


図6 スプールライブラリセーブの形式⁽³⁾

3-3-6 内部表現変換

データの内部表現変換は、取り出された有効データをACOSのデータの内部表現からFACOMのデータの内部表現に変換を行なうものである。変換は、4バイト整数、単精度実数、JIS文字のデータを変換できるようにした。ACOSでは、4バイト整数と単精度実数のデータの内部表現は、図7のようになっている。文字のデータは、JISコードで1ワードに4文字入

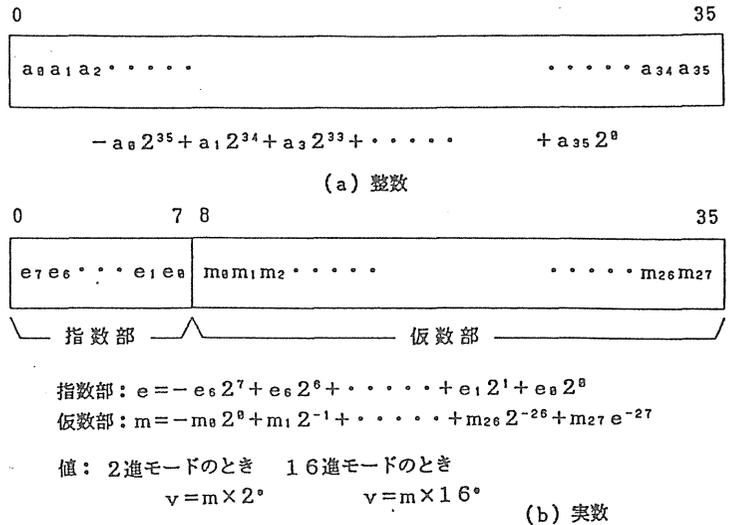
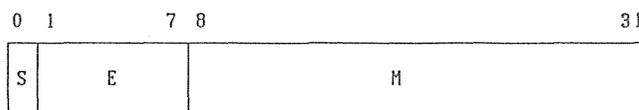


図7 ACOSのデータの内部表現⁽²⁾

っているものを処理することとした。FACOM⁽⁹⁾では、整数の表現は語長の違いだけでACOSと同様であり、実数は図8の様になっており、文字データの内部表現は、EBCDICコードで1ワードに4文字入っている。



S : 符号 (0のとき正, 1のとき負)
 E : 実際の指数に 2^6 を加えてある指数部
 M : ビット8の左側に小数点をもつ16進数の仮数部

図8 FACOMのデータの内部表現⁽⁴⁾

整数データの変換は、ACOSの36ビットから最上位の符号ビットと下位31ビット

を取り出し、FACOMの1ワードの中にビット操作⁽⁹⁾で入れる。語長の差の4ビットは、捨てるかたちになるので、数値の表現可能な値域は、 $-2^{31}(-2147483648)$ から $2^{31}-1(2147483647)$ になり、オーバーフロー、アンダーフローが発生した場合は、整数値の最大値または最小値を与えるようになっている。

単精度実数の変換は、ACOS36ビットから指数部の8ビットを仮数部の上位から24ビットを取り出し、ビット操作で指数部(e)と仮数部(m)の値を求め、 $v=m*2**e$ または、 $v=m*16**e$ の計算を行なってFACOMの1ワードの中に入れる。表現可能な値域は、仮数部が正の時、5.39760535E-79から 7.23700515E+75で、仮数部が負の時、-5.39760535E-79から -7.23700515E+75までで、オーバーフロー、アンダーフローが発生した場合は、実数値の最大値または最小値を与えるようになっている。

文字データの変換は、ACOSの36ビットから意味のある4バイト(32ビット)を取り出し、1バイトずつJISコードからEBCDICコードに変換して、FACOMの1ワードの中に入れる。

3-4 変換プログラムの実行

変換プログラムを実行する場合は、変換したいデータテープの形式に合わせて変換プログラムを選び、書式付きのデータならばそのまま実行すればよい。書式なしのデータの場合は、レコード数、各レコード、データ表現、データ数に合わせてプログラムを修正して、入力ファイル名、出力ファイル名、ブロック長、レコード長を指定して実行すればよい。書式なしデータの場合は、プログラムの修正が必要であるが、多く利用されている形式については、予め数例を用意しなるべく修正を必要としないようにしてある。

4. プログラム例と変換例

- | | |
|----------------------------------|----------------|
| (1) ブロック長、レコード長決定プログラム | プログラム名 GCMFI |
| (2) ユーティリティコピーの順編成分割レコード書式なし | プログラム名 FRCIBMO |
| (3) ユーティリティコピーの順編成可変長レコード書式付き | プログラム名 USVLRF |
| (4) スプールライブラリーセーブの順編成可変長レコード書式付き | プログラム名 SSVLRF |

上記の(1)から(4)のデータテープについてのプログラム例と変換例を付録に示し、以下に説明する。

(2)のデータテープの変換例は、1000個の整数データ(-499~500)と1000個の実数データ(-499.0~500.0)が入ってるデータテープを変換した例である。このプログラムの変換スピードは、FACOM(M-780)で、約100MB(メガバイト)(2400フィート磁

気テープ1巻分)の整数データを変換した時のスピードは約3分20秒で、実数データを変換した時のスピードは約3分45秒であった。

(3)の変換例は、文字データのデータテープを変換した例である。

(4)の変換例は、文字データのデータテープを変換した例である。2行目はACOS-6でのファイル名を示している。

5. まとめ

今回のプログラムを開発したことにより、FACOM側で直接ACOSのオリジナルデータテープを変換できる様になり、今までよりACOSからFACOMへのデータ交換が容易になった。変換スピードも実用に耐えられるスピードであり、今後利用価値があると思われる。今後の課題として、内部表現変換では、倍精度実数、4倍精度実数の変換ができるようにしたい。また今回作成しなかった形式についても変換プログラムを開発したい。

謝辞

この報告を書くにあたり御助力願った空電研究所の柿沼隆清教授、荻野龍樹助手、岩田晃助手ならびに、名古屋大学大型計算機センターの岡部直木助手に深く感謝します。開発にあたり、名古屋大学大型計算機センター(FACOM M-780)と、センターのライブラリーにある赤塚保雄氏作成のビット転送サブルーチンBITMVと、国立天文台豊川観測所のデータを、利用させて頂きました。

参考文献

- (1) オペレーティングシステムACOS-6/MVX データ管理 UFAS (V) 説明書, 日本電気
- (2) オペレーティングシステムACOS-6/MVX FORTRAN77 (V) のプログラミング手引書, 日本電気
- (3) オペレーティングシステムACOS-6 スプールシステム, 日本電気
- (4) オペレーティングシステムACOS-6 データ管理サービスプログラム (I) 説明書, 日本電気
- (5) 名古屋大学大型計算機センター 利用の手引, 昭和63年4月
- (6) 名古屋大学大型計算機センター データ・セット利用の手引, 昭和57年3月
- (7) FORTRAN 77 言語説明書, 日本電気
- (8) FACOM OS IV FORTRAN 77 文法書, 富士通
- (9) FACOM OS IV/F4 FORTRAN HE 使用手引書, 富士通

```
00010 C      FILE NAME  GCMFI
00020 C      CREATED BY   K. OHTA   11/29/88
00030 C      GCM FOR DATA CONVERSION ACOS(FRC) --> FACOM(IBM)
00040      DIMENSION NWD(100),NGCM(100)
00050      DATA NO/0/,IST/1/,NN/0/
00060 C
00070      2  CONTINUE
00080      NO=NO+1
00090      WRITE(6,*) NO,' BANME NO WRITE BUN NO WORD SUH',
00100      & ' OWARU TOKIWA 0'
00110      READ(5,*) NDN
00120 C
00130      IF(IST.EQ.1) THEN
00140      IF(NDN.EQ.0) GO TO 3
00150      IF(NDN+2.GE.320) THEN
00160      NN=NN+1
00170      NWD(NN)=NDN+2
00180      ND=0
00190      IST=1
00200      ELSE
00210      ND=NDN+2
00220      IST=2
00230      ENDIF
00240      GO TO 2
00250      ENDIF
00260 C
00270      IF(IST.EQ.2) THEN
00280      IF(NDN.EQ.0) THEN
00290      NN=NN+1
00300      NWD(NN)=ND
00310      GO TO 3
00320      ENDIF
00330 C
00340      IF(ND+NDN+1.GT.320) THEN
00350      NN=NN+1
00360      NWD(NN)=ND
00370      ND=NDN+2
00380      IST=2
00390 C
00400      IF(ND.GE.320) THEN
00410      NN=NN+1
00420      NWD(NN)=ND
00430      ND=0
00440      IST=1
00450      ENDIF
00460 C
00470      ELSE
00480      ND=ND+NDN+1
00490      IST=2
00500      ENDIF
00510      GO TO 2
00520      ENDIF
00530      3  CONTINUE
00540 C
00550      DO 10 J=1,NN
00560      IA=NWD(J)
00570      IB=MAXO(IA,IB)
00580      10  CONTINUE
00590      IBL=1440
00600      IF(IB.LT.320) IBL=INT(FLOAT(IB)*36.0/8.0+0.5)
00610 C
00620      IF(NN.EQ.1) CALL GCM1(NWD(1),LRL)
00630      IF(NN.GE.2) THEN
00640      DO 20 N1=1,NN
00650      ID1=NWD(N1)
00660      CALL GCM1(ID1,LRL)
```

```

00670      NGCM(N1)=LRL
00680      20  CONTINUE
00690 C
00700      DO 30 N1=1,NN-1
00710      ID1=NGCM(N1)
00720      ID2=NGCM(N1+1)
00730      CALL GCM2(ID1,ID2,LRL)
00740      NGCM(N1+1)=LRL
00750      30  CONTINUE
00760      ENDIF
00770 C
00780      WRITE(6,*) ' RECFM= FB  BLKSIZE=',IBL,'  LRECL=',LRL
00790      STOP
00800      END
00810      SUBROUTINE GCM1(ND1,NGC)
00820 C      GCM FOR DATA CONVERSION ACOS(FRC) --> FACOM(IBM)
00830      IF(ND1.LT.320) NGC=INT(FLOAT(ND1)*36.0/8.0+0.5)
00840      IF(ND1.GE.320) THEN
00850      ID1=1440
00860      ND2=MOD(ND1-3,318)+1
00870      ID2=INT(FLOAT(ND2+2)*36.0/8.0+0.5)
00880      CALL GCM2(ID1,ID2,NGC)
00890      ENDIF
00900      RETURN
00910      END
00920      SUBROUTINE GCM2(ID1,ID2,NGC)
00930      IF(ID1.GE.ID2) GO TO 2
00940      L1=ID2
00950      M1=ID1
00960      GO TO 3
00970      2  L1=ID1
00980      M1=ID2
00990      3  NN=L1-L1/M1*M1
01000      IF(NN.EQ.0) GO TO 4
01010      L1=M1
01020      M1=NN
01030      GO TO 3
01040      4  NGC=M1
01050      RETURN
01060      END

```

```
00010 //C42526A JOB REGION=1000K,CLASS=A
00020 //*
00030 //* *** ACOS(FRC) --> FACOM(IBM) *** ** FRCIBMO **
00040 //*
00050 //FORT77 EXEC FORTCLG,PARM='S,OPT(2)'
00060 //FORT.SYSIN DD DSN=C42526A.SUBA.FORT(SUBFRC2),DISP=SHR
00070 // DD *
00080 C FILE NAME FRCIBMO
00090 C CODED BY S. TAKATA 08/06/88
00100 C MODIFIED BY K. OHTA 09/28/88
00110 C DATA CONVERSION PROGRAM FROM ACOS(FRC) TO FACOM(IBM)
00120 PROGRAM FRCIBM
00130 PARAMETER (IBLOCK=1440,LRECL=9,NB=640)
00140 PARAMETER (ND1=1000,ND2=1000,N1=(ND1+ND2)*2)
00150 DIMENSION IB36B(NB),IRCB(N1),IDTA(ND1),RDTA(ND2)
00160 CHARACTER IDBUF(IBLOCK)*1
00170 C
00180 DO 1100 IL=1,2
00190 IFR=10+IL
00200 IFW=20+IL
00210 WRITE(6,100)
00220 100 FORMAT(1H )
00230 C
00240 IWC=0
00250 1 CONTINUE
00260 CALL RBLOCK(IFR,LRECL,IDBUF,IB36B,IBCWN,IBCWL,IEND)
00270 IF(IEND.EQ.1) GO TO 1100
00280 CALL SQRCW(IB36B,IRCWL,ICNT,ISGCD,IMDCD,IRPCD,ISGNO)
00290 C
00300 WRITE(6,200) IBCWN,IRCWL,ISGCD,IMDCD,IRPCD,ISGNO
00310 200 FORMAT(1H ,5X,'BCW COUNT=',I5,5X,'RCW LENGTH=',I6,
00320 & 5X,'SEGMENT CODE=',I3,5X,'MEDIA CODE=',I3,
00330 & 5X,'REPORT CODE=',I3,5X,'SEGMENT NUMBER=',I4)
00340 C
00350 DO 20 IWR=3,IRCWL+2
00360 IWC=IWC+2
00370 IRCB(IWC-1)=IB36B(IWR*2-1)
00380 IRCB(IWC)=IB36B(IWR*2)
00390 20 CONTINUE
00400 C
00410 IF(ISGCD.EQ.3.OR.ISGCD.EQ.0) GO TO 4
00420 GO TO 1
00430 4 CONTINUE
00440 C
00450 NW1=90
00460 C *** READ INTEGER READ REAL ***
00470 IF(IL.EQ.1) THEN
00480 IF(IWC/2.EQ.ND1) THEN
00490 IR1=1
00500 IRL=ND1
00510 CALL INTCNV(IRCB,IDTA,IR1,IRL,ICNT)
00520 WRITE(6,300)(IDTA(I),I=1,NW1)
00530 WRITE(6,300)(IDTA(I),I=ICNT-NW1+1,ICNT)
00540 300 FORMAT(2(2X,10I6))
00550 WRITE(IFW)(IDTA(I),I=1,ICNT)
00560 ENDIF
00570 C
00580 IF(IWC/2.EQ.ND1+ND2) THEN
00590 IR1=ND1+1
00600 IRL=ND1+ND2
00610 CALL REACNH(IRCB,RDTA,IR1,IRL,ICNT)
00620 WRITE(6,400)(RDTA(I),I=1,NW1)
00630 WRITE(6,400)(RDTA(I),I=ICNT-NW1+1,ICNT)
00640 400 FORMAT(2X,15F8.1)
00650 WRITE(IFW)(RDTA(I),I=1,ICNT)
00660 ENDIF
```

```
00670         ENDIF
00680 C
00690 C *** READ INTEGER REAL ***
00700         IF(IL.EQ.2) THEN
00710             IR1=1
00720             IRL=ND1
00730             CALL INTCNV(IRCB,IDTA,IR1,IRL,ICNT)
00740             WRITE(6,300)(IDTA(I),I=1,NW1)
00750             WRITE(6,300)(IDTA(I),I=ICNT-NW1+1,ICNT)
00760 C
00770             IR1=ND1+1
00780             IRL=ND1+ND2
00790             CALL REACNH(IRCB,RDTA,IR1,IRL,ICNT)
00800             WRITE(6,400)(RDTA(I),I=1,NW1)
00810             WRITE(6,400)(RDTA(I),I=ICNT-NW1+1,ICNT)
00820             WRITE(IFW)(IDTA(I),I=1,ICNT),(RDTA(I),I=1,ICNT)
00830             ENDIF
00840 C
00850             IF(ISGCD.EQ.3.OR.ISGCD.EQ.0) GO TO 1
00860 1100 CONTINUE
00870         STOP
00880         END
00890 //GO.FT11F001 DD DISP=SHR,DSN=A41456A.INTREA1
00900 //GO.FT12F001 DD DISP=SHR,DSN=A41456A.INTREA2
00910 //GO.FT21F001 DD DISP=(NEW,CATLG),DSN=C42526A.IBMIR1.DATA,
00920 //             UNIT=PUB,SPACE=(TRK,(10,5),RLSE)
00930 //GO.FT22F001 DD DISP=(NEW,CATLG),DSN=C42526A.IBMIR2.DATA,
00940 //             UNIT=PUB,SPACE=(TRK,(10,5),RLSE)
00950 //
```

サブルーチンプログラム

```

00010 C      FILE NAME  SUBA(SUBFRC3)
00020 C      CODED BY   S. TAKATA   08/06/88
00030 C      MODIFIED BY K. OHTA    11/25/88
00040 C      DATA CONVERSION PROGRAM FROM(FRC) TO FACOM(IBM)
00050      SUBROUTINE RBLOCK(IFL,LRECL,IDBUF,IB36B,IBCWN,IBCWL,IEND)
00060      DIMENSION IB36B(1)
00070      CHARACTER IDBUF(1)*1,FRMT*24
00080 C
00090      CALL MKFMT(LRECL,FRMT)
00100      IEND=0
00110      IF(LRECL.LT.5) THEN
00120      IF(LRECL.EQ.1) KREC=5/LRECL
00130      IF(LRECL.NE.1) KREC=5/LRECL+1
00140      JRECL=KREC*LRECL
00150      DO 10 IL=1,KREC
00160      I1=LRECL*(IL-1)+1
00170      I2=LRECL*IL
00180      READ(IFL,FRMT,END=97,ERR=99)(IDBUF(I),I=I1,I2)
00190      10 CONTINUE
00200      ELSE
00210      JRECL=LRECL
00220      READ(IFL,FRMT,END=97,ERR=99)(IDBUF(I),I=1,LRECL)
00230      ENDIF
00240 C
00250      CALL BIT36C(IDBUF,LRECL,IB36B)
00260      IBCWN=0
00270      IBCWL=0
00280      CALL BITMV(IB36B(1),0,18,IBCWN,14)
00290      CALL BITMV(IB36B(1),18,14,IBCWL,14)
00300      CALL BITMV(IB36B(2),0,4,IBCWL,28)
00310      JBSZ=(FLOAT((IBCWL+1)*4)/32.0+FLOAT(IBCWL+1))*4.0+0.5
00320 C
00330      IF(JBSZ.NE.LRECL) THEN
00340      DO 20 IREC=JRECL+1,JBSZ,LRECL
00350      I1=IREC
00360      I2=IREC+LRECL-1
00370      READ(IFL,FRMT,END=98,ERR=99)(IDBUF(I),I=I1,I2)
00380      20 CONTINUE
00390      GO TO 2
00400      98 CONTINUE
00410      JBSZ=IREC-1
00420      IEND=2
00430      2 CONTINUE
00440      CALL BIT36C(IDBUF,JBSZ,IB36B)
00450      ENDIF
00460 C
00470      RETURN
00480      97 CONTINUE
00490      IEND=1
00500      RETURN
00510      99 CONTINUE
00520      WRITE(6,*) ' *** READ DATA ERROR ***'
00530      RETURN
00540      END
00550      SUBROUTINE MKFMT(LRECL,FRMT)
00560      CHARACTER FMTN*24,FRMT*24
00570 C
00580      IF(LRECL.GT.250) THEN
00590      LR1=LRECL/250
00600      LR2=MOD(LRECL,250)
00610      IF(LR2.EQ.0) THEN
00620      WRITE(FMTN,100) LR1
00630      100 FORMAT('(',I4,'(250A1)')
00640      ELSE
00650      WRITE(FMTN,200) LR1,LR2
00660      200 FORMAT('(',I4,'(250A1)',',',I4,'A1)')

```

```

00670         ENDIF
00680         ELSE
00690         WRITE(FMTN,300) LRECL
00700 300      FORMAT('(',I5,'A1')
00710         ENDIF
00720 C
00730         FRMT=' '
00740         JCNT=0
00750         DO 10 ICNT=1,24
00760         IF(FMTN(ICNT:ICNT).NE.' ') THEN
00770         JCNT=JCNT+1
00780         FRMT(JCNT:JCNT)=FMTN(ICNT:ICNT)
00790         ENDIF
00800 10      CONTINUE
00810         RETURN
00820         END
00830         SUBROUTINE BIT36C(IDBUF,IBSZ,IB36B)
00840         DIMENSION IB36B(1)
00850         CHARACTER IDBUF(1)*1
00860 C
00870         N=0
00880         DO 10 IBLK=1,IBSZ,9
00890 C         *** 1 WORD ***
00900         N=N+2
00910         N1=N-1
00920         CALL BITMV(IDBUF(IBLK),0,8,IB36B(N1),0)
00930         CALL BITMV(IDBUF(IBLK+1),0,8,IB36B(N1),8)
00940         CALL BITMV(IDBUF(IBLK+2),0,8,IB36B(N1),16)
00950         CALL BITMV(IDBUF(IBLK+3),0,8,IB36B(N1),24)
00960         CALL BITMV(IDBUF(IBLK+4),0,4,IB36B(N),0)
00970 C         *** 2 WORD ***
00980         N=N+2
00990         N1=N-1
01000         CALL BITMV(IDBUF(IBLK+4),4,4,IB36B(N1),0)
01010         CALL BITMV(IDBUF(IBLK+5),0,8,IB36B(N1),4)
01020         CALL BITMV(IDBUF(IBLK+6),0,8,IB36B(N1),12)
01030         CALL BITMV(IDBUF(IBLK+7),0,8,IB36B(N1),20)
01040         CALL BITMV(IDBUF(IBLK+8),0,4,IB36B(N1),28)
01050         CALL BITMV(IDBUF(IBLK+8),4,4,IB36B(N),0)
01060 10      CONTINUE
01070         RETURN
01080         END
01090         SUBROUTINE SQRCW(IB36B,IRCWL,ICCNT,ISGCD,IMDCD,IRPCD,ISGNO)
01100         DIMENSION IB36B(1)
01110 C
01120         IRCWL=0
01130         CALL BITMV(IB36B(3),0,18,IRCWL,14)
01140 C
01150         ICCNT=0
01160         CALL BITMV(IB36B(3),18,2,ICCNT,30)
01170         ISGCD=0
01180         CALL BITMV(IB36B(3),24,2,ISGCD,30)
01190 C
01200         IMDCD=0
01210         IRPCD=0
01220         ISGNO=1
01230         IF(ISGCD.EQ.1) CALL BITMV(IB36B(3),26,4,IMDCD,28)
01240         IF(ISGCD.EQ.1) CALL BITMV(IB36B(3),30,2,IRPCD,26)
01250         IF(ISGCD.EQ.1) CALL BITMV(IB36B(4),0,4,IRPCD,28)
01260         IF(ISGCD.GE.2) CALL BITMV(IB36B(3),26,6,ISGNO,22)
01270         IF(ISGCD.GE.2) CALL BITMV(IB36B(4),0,4,ISGNO,28)
01280         RETURN
01290         END
01300         SUBROUTINE INTCNV(IRCB,IDTA,IR1,IRL,ICNT)
01310         DIMENSION IRCB(1),IDTA(1)
01320 C

```

```

01330      INO=0
01340      DO 30 IRNO=IR1,IRL
01350      INO=INO+1
01360      IWH=IRCB(IRNO*2-1)
01370      IWL=IRCB(IRNO*2)
01380      CALL BITMV(IWH,0,1,IDAT,0)
01390      CALL BITMV(IWH,5,27,IDAT,1)
01400      CALL BITMV(IWL,0,4,IDAT,28)
01410 C
01420      CALL BITMV(IWH,0,5,IOV,27)
01430      IF(IOV.GE.1.AND.IOV.LE.15) IDAT=2147483647
01440      IDTA(INO)=IDAT
01450 30 CONTINUE
01460      ICNT=INO
01470      RETURN
01480      END
01490      SUBROUTINE REACNB(IRCB,RDTA,IR1,IRL,ICNT)
01500      PARAMETER (KP=127,KM=128,KE=KP+KM+1,IA2=2**30)
01510      DIMENSION IRCB(1),RDTA(1),BTBL(KE)
01520      EQUIVALENCE (IA,AM)
01530 C
01540      DO 10 KNO=1,KE
01550      BTBL(KNO)=2.0**(KNO-KM-1)
01560 10 CONTINUE
01570 C
01580      INO=0
01590      DO 30 IRNO=IR1,IRL
01600      INO=INO+1
01610      IWH=IRCB(IRNO*2-1)
01620      IWL=IRCB(IRNO*2)
01630 C
01640      ME=0
01650      CALL BITMV(IWH,0,1,ME,31)
01660      IE=0
01670 C      *** IE ONE WORD ALL BIT 1 ***
01680      IF(ME.EQ.1) IE=-1
01690      CALL BITMV(IWH,1,7,IE,25)
01700      IF(IE.GE.KP) IE=KP
01710      IF(IE.LE.-KM) IE=-KM
01720      JE=IE+KM+1
01730 C
01740      MM=0
01750      CALL BITMV(IWH,8,1,MM,31)
01760 C      *** SISUH NI 64 ZYO KUWAERU ***
01770      IA=IA2
01780      CALL BITMV(IWH,9,23,AM,8)
01790      IF(MM.EQ.1) AM=AM-1.0
01800      RDTA(INO)=AM*BTBL(JE)
01810 30 CONTINUE
01820      ICNT=INO
01830      RETURN
01840      END
01850      SUBROUTINE REACNH(IRCB,RDTA,IR1,IRL,ICNT)
01860      PARAMETER (KP=62,KM=64,KE=KP+KM+1,IA2=2**30)
01870      DIMENSION IRCB(1),RDTA(1),HTBL(KE)
01880      EQUIVALENCE (IA,AM)
01890 C
01900      DO 10 KNO=1,KE
01910      HTBL(KNO)=16.0**(KNO-KM-1)
01920 10 CONTINUE
01930 C
01940      INO=0
01950      DO 30 IRNO=IR1,IRL
01960      INO=INO+1
01970      IWH=IRCB(IRNO*2-1)
01980      IWL=IRCB(IRNO*2)

```

```
01990 C
02000 ME=0
02010 CALL BITMV(IWH,0,1,ME,31)
02020 IE=0
02030 C *** IE ONE WORD ALL BIT 1 ***
02040 IF(ME.EQ.1) IE=-1
02050 CALL BITMV(IWH,1,7,IE,25)
02060 IF(IE.GE.KP) IE=KP
02070 IF(IE.LE.-KM) IE=-KM
02080 JE=IE+KM+1
02090 C
02100 MM=0
02110 CALL BITMV(IWH,8,1,MM,31)
02120 C *** SISUH NI 64 ZYOH KUWAERU ***
02130 IA=IA2
02140 CALL BITMV(IWH,9,23,AM,8)
02150 IF(MM.EQ.1) AM=AM-1.0
02160 RDTA(INO)=AM*HTBL(JE)
02170 30 CONTINUE
02180 ICNT=INO
02190 RETURN
02200 END
```

```

00010 //C42567A JOB REGION=2000K,CLASS=B
00020 //FORT77 EXEC FORTCLG,PARM.FORT='SOURCE,OPT(2) '
00030 //FORT.SYSIN DD DSN=C42567A.AFCLIBF2.FORT,DISP=SHR
00040 // DD *
00050 C FILE NAME USVLRF
00060 C CODED BY S.TAKATA 09/01/88 (THURSDAY)
00070 C
00080 PROGRAM USVLRF
00090 C
00100 PARAMETER (IUN1=12,IUN2=13,LS=0)
00110 PARAMETER (NDP1=10000,NDP2=4000,IBSZ=1440,LRECL=1)
00120 DIMENSION IB36B(NDP1)
00130 CHARACTER IDBUF(NDP1*2)*1,DATA(NDP2)*1,JECHR*4
00140 CHARACTER INFL*20,OTFL*20,FRMT*30
00150 DATA INFL/'BPTBUF'/,OTFL/'DPTBUF'/
00160 C
00170 C *** OPEN FILE ***
00180 WRITE(6,100)
00190 100 FORMAT(1H1/' *** USFVLR DATA *** '/')
00200 OPEN(IUN1,IOSTAT=1ST,ERR=999,FILE=INFL,STATUS='OLD',
00210 & ACCESS='SEQUENTIAL',FORM='FORMATTED')
00220 OPEN(IUN2,IOSTAT=1ST,ERR=999,FILE=OTFL,STATUS='NEW',
00230 & ACCESS='SEQUENTIAL',FORM='FORMATTED')
00240 C
00250 C *** READ ONE BLOCK AND CHECK BCW,RCW ***
00260 10 CONTINUE
00270 CALL RBLOCK(IUN1,IBSZ,LS,LRECL,IDBUF,IB36B,JBSZ,IBCWN,IBCWL,
00280 & IEND)
00290 IF(IEND.EQ.1)GO TO 50
00300 IWNO=2
00310 70 CONTINUE
00320 CALL RCW(IB36B,IWNO,IRCWL,ICCNT,IMDCD,IRPCD,ISGCD)
00330 IF(IMDCD.EQ.8)THEN
00340 CALL BITMVI(IB36B((IWNO+1)*2-1),28,4,IUBLK,14)
00350 CALL BITMVI(IB36B((IWNO+1)*2),0,14,IUBLK,18)
00360 WRITE(6,200)IUBLK,IUBLK
00370 200 FORMAT(' AVAILABLE BLOCK ',I5,1X,Z8)
00380 ENDIF
00390 C
00400 C *** CONVERSION JIS TO EBCDIC AND WRITE DATA ***
00410 IF(IMDCD.EQ.6)THEN
00420 DO 80 IBFC=1,4000
00430 DATA(IBFC)=' '
00440 80 CONTINUE
00450 ICNT=0
00460 JCNT=0
00470 DO 90 JWDNO=IWNO+1,IWNO+IRCWL
00480 ICNT=ICNT+1
00490 IWDNO=JWDNO
00500 CALL JECONV(IB36B,IWDNO,JECHR)
00510 IF(ICNT.NE.IRCWL)THEN
00520 JCH=4
00530 ELSE
00540 IF(ICCNT.EQ.0)JCH=4
00550 IF(ICCNT.EQ.1)JCH=1
00560 IF(ICCNT.EQ.2)JCH=2
00570 IF(ICCNT.EQ.3)JCH=3
00580 ENDIF
00590 DO 110 ICH=1,JCH
00600 JCNT=JCNT+1
00610 DATA(JCNT)=JECHR(ICH:ICH)
00620 110 CONTINUE
00630 90 CONTINUE
00640 CALL MKFMT(1,FRMT)
00650 WRITE(IUN2,FRMT)(DATA(IF),IF=1,JCNT)
00660 WRITE(6,*)(DATA(IP),IP=1,JCNT)

```

```
00670             ENDIF
00680             IWNO=IWNO+IRCWL+1
00690             IF(IWNO.LE.IBCWL)GO TO 70
00700             IF(IEND.EQ.2)GO TO 50
00710             GO TO 10
00720 C
00730 C *** CLOSE FILE ***
00740     50 CONTINUE
00750             CLOSE(IUN1,IOSTAT=IST,ERR=999,STATUS='KEEP')
00760             CLOSE(IUN2,IOSTAT=IST,ERR=999,STATUS='KEEP')
00770             GO TO 60
00780     999 CONTINUE
00790             WRITE(6,*)'### OPEN OR CLOSE ERROR (SUBROUTINE USFVLR) ###'
00800     60 CONTINUE
00810             STOP
00820             END
00830 //
```

```
00010 //C42567A JOB REGION=2000K,CLASS=B
00020 //FORT77 EXEC FORTCLG,PARM.FORT='SOURCE,OPT(2)'
00030 //FORT.SYSIN DD DSN=C42567A.AFCLIBF2.FORT,DISP=SHR
00040 // DD *
00050 C FILE NAME SSVLRF
00060 C CODED BY S.TAKATA 09/01/88 (THURSDAY)
00070 C
00080 PROGRAM SSVLRF
00090 C
00100 PARAMETER (IUN1=12,IUN2=13,LS=0)
00110 PARAMETER (NDP1=10000,IBSZ=7000,LRECL=7)
00120 DIMENSION IB36B(NDP1)
00130 CHARACTER IDBUF(NDP1*2)*1,ADATA(NDP1)*4,BDATA(NDP1*4)*1,JECHR*4
00140 CHARACTER INFL*20,OTFL*20,FRMT*30
00150 DATA INFL/'BPTBUF'/,OTFL/'DPTBUF'/
00160 C
00170 C *** OPEN FILE ***
00180 WRITE(6,100)
00190 100 FORMAT(1H1/' *** SSVLR DATA *** '/')
00200 OPEN(IUN1,IOSTAT=IST,ERR=999,FILE=INFL,STATUS='OLD',
00210 & ACCESS='SEQUENTIAL',FORM='FORMATTED')
00220 OPEN(IUN2,IOSTAT=IST,ERR=999,FILE=OTFL,STATUS='NEW',
00230 & ACCESS='SEQUENTIAL',FORM='FORMATTED')
00240 C
00250 C *** READ ONE BLOCK AND CHECK BCW,RCW ***
00260 10 CONTINUE
00270 CALL RBLOCK(IUN1,IBSZ,LS,LRECL,IDBUF,IB36B,JBSZ,IBCWN,IBCWL,
00280 & IEND)
00290 IBCWNP=IBCWN
00300 IBCWLP=IBCWL
00310 IF(IBCWNP.EQ.2)GO TO 60
00320 IF(IEND.EQ.1)GO TO 50
00330 DO 40 IBLK=2,IBCWLP+1,321
00340 IWNO=IBLK
00350 CALL RCW(IB36B,IWNO,IRCWL,ICCNT,IMDCD,IRPCD,ISGCD)
00360 C
00370 C *** CHECK SPOOL LIBRARY SAVE FILE NAME ***
00380 IF(IBCWNP.EQ.1.AND.IBLK.EQ.2)THEN
00390 DO 70 IBFC=1,10000
00400 ADATA(IBFC)=' '
00410 70 CONTINUE
00420 ICNT=0
00430 DO 80 JWDNO=IBLK+1,IBLK+IRCWL+1
00440 ICNT=ICNT+1
00450 IWDNO=JWDNO
00460 CALL JECONV(IB36B,IWDNO,JECHR)
00470 ADATA(ICNT)=JECHR
00480 80 CONTINUE
00490 WRITE(6,*)(ADATA(IP),IP=1,ICNT)
00500 ELSE
00510 IWNO=IBLK+1
00520 CALL BCW(IB36B,IWNO,IBCWN,IBCWL)
00530 IRNO=IBLK+2
00540 90 CONTINUE
00550 IWNO=IRNO
00560 CALL RCW(IB36B,IWNO,IRCWL,ICCNT,IMDCD,IRPCD,ISGCD)
00570 IF(IRCWL.EQ.0)GO TO 50
00580 IF(IMDCD.EQ.8)THEN
00590 CALL BITMVI(IB36B((IWNO+1)*2-1),28,4,IUBLK,14)
00600 CALL BITMVI(IB36B((IWNO+1)*2),0,14,IUBLK,18)
00610 WRITE(6,200)IUBLK,IUBLK
00620 200 FORMAT(' AVAILABLE BLOCK ',I5,1X,Z8)
00630 ENDIF
00640 C
00650 C *** CONVERSION JIS TO EBCDIC AND WRITE DATA ***
00660 IF(IMDCD.EQ.6)THEN
```

```

00670          DO 110 IBFC=1,40000
00680          BDATA(IBFC)=' '
00690 110      CONTINUE
00700          ICNT=0
00710          JCNT=0
00720          DO 120 JWDNO=IRNO+1,IRNO+IRCWL
00730          ICNT=ICNT+1
00740          IWDNO=JWDNO
00750          CALL JECONV(1B36B,IWDNO,JECHR)
00760          IF(ICNT.NE.IRCWL)THEN
00770          JCH=4
00780          ELSE
00790          IF(ICCNT.EQ.0)JCH=4
00800          IF(ICCNT.EQ.1)JCH=1
00810          IF(ICCNT.EQ.2)JCH=2
00820          IF(ICCNT.EQ.3)JCH=3
00830          ENDIF
00840          DO 130 ICH=1,JCH
00850          JCNT=JCNT+1
00860          BDATA(JCNT)=JECHR(ICH:ICH)
00870 130      CONTINUE
00880 120      CONTINUE
00890          CALL MKFMT(1,FRMT)
00900          WRITE(IUN2,FRMT)(BDATA(IF),IF=1,JCNT)
00910          ENDIF
00920          WRITE(6,*)(BDATA(IP),IP=1,JCNT)
00930          IRNO=IRNO+IRCWL+1
00940          IF(IRNO-IBLK-2.LT.IBCWL)GO TO 90
00950          ENDIF
00960 40      CONTINUE
00970          IF(IEND.EQ.2)GO TO 50
00980          GO TO 10
00990 C
01000 C *** CLOSE FILE ***
01010 50 CONTINUE
01020          CLOSE(IUN1,Iostat=IST,ERR=999,STATUS='KEEP')
01030          CLOSE(IUN2,Iostat=IST,ERR=999,STATUS='KEEP')
01040          GO TO 60
01050 999 CONTINUE
01060          WRITE(6,*)'### OPEN OR CLOSE ERROR (SUBROUTINE SSFVLR) ###'
01070 60 CONTINUE
01080          STOP
01090          END
01100 //

```

例5 FRCIBMOの変換例

BCW COUNT=	1	RCW LENGTH=	310	SEGMENT CODE=	1	MEDIA CODE=	0	REPORT CODE=	12	SEGMENT NUMBER=	1								
BCW COUNT=	2	RCW LENGTH=	310	SEGMENT CODE=	2	MEDIA CODE=	0	REPORT CODE=	0	SEGMENT NUMBER=	2								
BCW COUNT=	3	RCW LENGTH=	310	SEGMENT CODE=	2	MEDIA CODE=	0	REPORT CODE=	0	SEGMENT NUMBER=	3								
BCW COUNT=	4	RCW LENGTH=	46	SEGMENT CODE=	3	MEDIA CODE=	0	REPORT CODE=	0	SEGMENT NUMBER=	4								
-499	-470	-497	-496	-495	-494	-493	-492	-491	-490	-489	-488	-487	-486	-485	-484	-483	-482	-481	-480
-479	-478	-477	-476	-475	-474	-473	-472	-471	-470	-469	-468	-467	-466	-465	-464	-463	-462	-461	-460
-459	-458	-457	-456	-455	-454	-453	-452	-451	-450	-449	-448	-447	-446	-445	-444	-443	-442	-441	-440
-439	-438	-437	-436	-435	-434	-433	-432	-431	-430	-429	-428	-427	-426	-425	-424	-423	-422	-421	-420
-419	-418	-417	-416	-415	-414	-413	-412	-411	-410										
411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430
431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450
451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470
471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490
491	492	493	494	495	496	497	498	499	500										
BCW COUNT=	5	RCW LENGTH=	310	SEGMENT CODE=	1	MEDIA CODE=	0	REPORT CODE=	12	SEGMENT NUMBER=	1								
BCW COUNT=	6	RCW LENGTH=	310	SEGMENT CODE=	2	MEDIA CODE=	0	REPORT CODE=	0	SEGMENT NUMBER=	2								
BCW COUNT=	7	RCW LENGTH=	310	SEGMENT CODE=	2	MEDIA CODE=	0	REPORT CODE=	0	SEGMENT NUMBER=	3								
BCW COUNT=	8	RCW LENGTH=	46	SEGMENT CODE=	3	MEDIA CODE=	0	REPORT CODE=	0	SEGMENT NUMBER=	4								
-499.0	-498.0	-497.0	-496.0	-495.0	-494.0	-493.0	-492.0	-491.0	-490.0	-489.0	-488.0	-487.0	-486.0	-485.0	-484.0	-483.0	-482.0	-481.0	-480.0
-480.0	-480.0	-482.0	-481.0	-480.0	-479.0	-478.0	-477.0	-476.0	-475.0	-474.0	-473.0	-472.0	-471.0	-470.0					
-469.0	-468.0	-467.0	-466.0	-465.0	-464.0	-463.0	-462.0	-461.0	-460.0	-459.0	-458.0	-457.0	-456.0	-455.0					
-454.0	-453.0	-452.0	-451.0	-450.0	-449.0	-448.0	-447.0	-446.0	-445.0	-444.0	-443.0	-442.0	-441.0	-440.0					
-439.0	-438.0	-437.0	-436.0	-435.0	-434.0	-433.0	-432.0	-431.0	-430.0	-429.0	-428.0	-427.0	-426.0	-425.0					
-424.0	-423.0	-422.0	-421.0	-420.0	-419.0	-418.0	-417.0	-416.0	-415.0	-414.0	-413.0	-412.0	-411.0	-410.0					
411.0	412.0	413.0	414.0	415.0	416.0	417.0	418.0	419.0	420.0	421.0	422.0	423.0	424.0	425.0					
426.0	427.0	428.0	429.0	430.0	431.0	432.0	433.0	434.0	435.0	436.0	437.0	438.0	439.0	440.0					
441.0	442.0	443.0	444.0	445.0	446.0	447.0	448.0	449.0	450.0	451.0	452.0	453.0	454.0	455.0					
456.0	457.0	458.0	459.0	460.0	461.0	462.0	463.0	464.0	465.0	466.0	467.0	468.0	469.0	470.0					
471.0	472.0	473.0	474.0	475.0	476.0	477.0	478.0	479.0	480.0	481.0	482.0	483.0	484.0	485.0					
486.0	487.0	488.0	489.0	490.0	491.0	492.0	493.0	494.0	495.0	496.0	497.0	498.0	499.0	500.0					
BCW COUNT=	1	RCW LENGTH=	310	SEGMENT CODE=	1	MEDIA CODE=	0	REPORT CODE=	13	SEGMENT NUMBER=	1								
BCW COUNT=	2	RCW LENGTH=	310	SEGMENT CODE=	2	MEDIA CODE=	0	REPORT CODE=	0	SEGMENT NUMBER=	2								
BCW COUNT=	3	RCW LENGTH=	310	SEGMENT CODE=	2	MEDIA CODE=	0	REPORT CODE=	0	SEGMENT NUMBER=	3								
BCW COUNT=	4	RCW LENGTH=	310	SEGMENT CODE=	2	MEDIA CODE=	0	REPORT CODE=	0	SEGMENT NUMBER=	4								
BCW COUNT=	5	RCW LENGTH=	310	SEGMENT CODE=	2	MEDIA CODE=	0	REPORT CODE=	0	SEGMENT NUMBER=	5								
BCW COUNT=	6	RCW LENGTH=	310	SEGMENT CODE=	2	MEDIA CODE=	0	REPORT CODE=	0	SEGMENT NUMBER=	6								
BCW COUNT=	7	RCW LENGTH=	92	SEGMENT CODE=	3	MEDIA CODE=	0	REPORT CODE=	0	SEGMENT NUMBER=	7								
-499	-498	-497	-496	-495	-494	-493	-492	-491	-490	-489	-488	-487	-486	-485	-484	-483	-482	-481	-480
-479	-478	-477	-476	-475	-474	-473	-472	-471	-470	-469	-468	-467	-466	-465	-464	-463	-462	-461	-460
-459	-458	-457	-456	-455	-454	-453	-452	-451	-450	-449	-448	-447	-446	-445	-444	-443	-442	-441	-440
-439	-438	-437	-436	-435	-434	-433	-432	-431	-430	-429	-428	-427	-426	-425	-424	-423	-422	-421	-420
-419	-418	-417	-416	-415	-414	-413	-412	-411	-410										
411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430
431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450
451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470
471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490
491	492	493	494	495	496	497	498	499	500										
-499.0	-498.0	-497.0	-496.0	-495.0	-494.0	-493.0	-492.0	-491.0	-490.0	-489.0	-488.0	-487.0	-486.0	-485.0	-484.0	-483.0	-482.0	-481.0	-480.0
-480.0	-480.0	-482.0	-481.0	-480.0	-479.0	-478.0	-477.0	-476.0	-475.0	-474.0	-473.0	-472.0	-471.0	-470.0					
-469.0	-468.0	-467.0	-466.0	-465.0	-464.0	-463.0	-462.0	-461.0	-460.0	-459.0	-458.0	-457.0	-456.0	-455.0					
-454.0	-453.0	-452.0	-451.0	-450.0	-449.0	-448.0	-447.0	-446.0	-445.0	-444.0	-443.0	-442.0	-441.0	-440.0					
-439.0	-438.0	-437.0	-436.0	-435.0	-434.0	-433.0	-432.0	-431.0	-430.0	-429.0	-428.0	-427.0	-426.0	-425.0					
-424.0	-423.0	-422.0	-421.0	-420.0	-419.0	-418.0	-417.0	-416.0	-415.0	-414.0	-413.0	-412.0	-411.0	-410.0					
411.0	412.0	413.0	414.0	415.0	416.0	417.0	418.0	419.0	420.0	421.0	422.0	423.0	424.0	425.0					
426.0	427.0	428.0	429.0	430.0	431.0	432.0	433.0	434.0	435.0	436.0	437.0	438.0	439.0	440.0					
441.0	442.0	443.0	444.0	445.0	446.0	447.0	448.0	449.0	450.0	451.0	452.0	453.0	454.0	455.0					
456.0	457.0	458.0	459.0	460.0	461.0	462.0	463.0	464.0	465.0	466.0	467.0	468.0	469.0	470.0					
471.0	472.0	473.0	474.0	475.0	476.0	477.0	478.0	479.0	480.0	481.0	482.0	483.0	484.0	485.0					
486.0	487.0	488.0	489.0	490.0	491.0	492.0	493.0	494.0	495.0	496.0	497.0	498.0	499.0	500.0					

USVLRFの変換例

```

*** USFVLR DATA ***
ZCZC 01/AUG 12:57
***** RCVD MSG ****
R2826
SSSS
2832612 DENPA J
4322310 TYKW J
01 1249

GEOALERT WWA214 010330Z
92431 11621 21870 30100 49980 50100
33222 01000 44512 01000 43015 04000 12327 06010 40125 00000
23925 01000
PRESTOS NONE
80001 77773 33222 ERUPTIVE 43015 ERUPTIVE 12327 ACTIVE
THREE QUIET REGIONS
SOLALERT 01/XX
MAGALERT MINOR 01/XX
SMN OBS 01 AUG ? FMS 1330:05 LMS 1428:12 NEXT FMS 1504:05.
SAAS 1339?20= 1714?25= 1851?26= 2028?27= 2208?25= 2348?22=
0128?20= 0308?15= 0447?04. SMN POINTING (N30E18 AT 01/1330UT
FOR 01/1330 ? 02/1358UT RS FW FC). SMN CONTINUES TRACKING
AR 5092. XRP RAN CRYSTAL ON H FLARE AT 31/0746UT.

SOLTERWARN

```

SSVLRFの変換例

```

*** SSFVLR DATA ***
3R1A/TX/PTBUF
RRR Y 1 1 Q8 Y F
AVAILABLE BLOCK 443 000001BB
ZCZC 01/AUG 12:57
***** RCVD MSG ****
R2826
SSSS
2832612 DENPA J
4322310 TYKW J
01 1249

GEOALERT WWA214 010330Z
92431 11621 21870 30100 49980 50100
33222 01000 44512 01000 43015 04000 12327 06010 40125 00000
23925 01000
PRESTOS NONE
80001 77773 33222 ERUPTIVE 43015 ERUPTIVE 12327 ACTIVE
THREE QUIET REGIONS
SOLALERT 01/XX
MAGALERT MINOR 01/XX
SMN OBS 01 AUG ? FMS 1330:05 LMS 1428:12 NEXT FMS 1504:05.
SAAS 1339?20= 1714?25= 1851?26= 2028?27= 2208?25= 2348?22=
0128?20= 0308?15= 0447?04. SMN POINTING (N30E18 AT 01/1330UT
FOR 01/1330 ? 02/1358UT RS FW FC). SMN CONTINUES TRACKING
AR 5092. XRP RAN CRYSTAL ON H FLARE AT 31/0746UT.

SOLTERWARN

```

